Amendments to the Claims:

This listing of claims replaces all prior versions and listings of claims in the application:

Listing of Claims:

1. (currently amended) A method for searching for one or more logical elements in a hierarchical tree structure of an ~~extended~~extensible markup language (XML) document conforming to a schema used for XML, comprising:

providing a representation of an ~~extended markup language~~XML document instance containing two or more logical elements, wherein at least one logical element is a parent node and at least one logical element is a child node in a hierarchical tree structure describing the representation;

receiving a query for logical elements satisfying an [[math]] XPath expression; [[and]]

searching in the hierarchical tree structure only nodes that potentially have child ~~10~~ ~~nodes~~nodes satisfying the XPath expression; and

providing the logical elements satisfying the XPath expression.

2. (currently amended) The method of claim 1, including the further step of generating a collection of parent nodes that potentially have child nodes satisfying the XPath expression from a table relating a class of parent nodes and a class of child nodes, and wherein the table is used in the ~~final 15~~ searching step.

3. (currently amended) The method of claim 1, including the further step of generating a collection of parent nodes that potentially have child nodes satisfying the [[math]]XPath expression from a table relating parent nodes and child nodes, and wherein the table is used in the ~~final~~ searching step.

4. (original) The method of claim 2, wherein the table comprises entries containing hash representations of a class of parent nodes and a class of child nodes.

5. (currently amended) The method of claim 3, wherein the table comprises entries containing hash [[25]] representations of the parent nodes and child nodes.

6. (original) The method of claim 2, wherein the table comprises a listing of permitted classes of child nodes for each class of parent node.

7. (original) The method of claim 3, wherein the table comprises a listing of child nodes for each parent node.

8. (original) The method of claim 2, wherein the table comprises a listing of permitted classes of parent nodes for each class of child node.

9. (original) The method of claim 3, wherein table comprises a listing of permitted parent nodes for each child node.

10. (currently amended) The method of claim 1, further comprising:

receiving a rule set identifying allowable combinations between child nodes and parent nodes in a hierarchical document structure;

transforming the rule set into a table relating a class of parent nodes and a class of child nodes; and

using the table in the [[final]] searching step.

11. (currently amended) The method of claim 1, further comprising:

receiving a rule set identifying allowable combinations between child nodes and parent nodes in a hierarchical document structure;

transforming the rule set into a table relating parent nodes and child nodes; and

using the table in the [[final]] searching step.

12. (original) The method of claim 10, wherein:

the rule set includes one of: an XML schema, a DTD, and a RelaxNg schema.

13. (original) The method of claim 11, wherein:

the rule set includes one of: an XML schema, a DTD, and a RelaxNg schema.

14. (original) The method of claim 2, wherein the table includes a listing of a not-permitted class of child nodes for each class of parent node.

15. (original) The method of claim 3, wherein the table includes a listing of not-permitted child nodes for each parent node.

16. (original) The method of claim 2, wherein the table includes a listing of a not-permitted class of parent nodes for each class of child node.

17. (original) The method of claim 3, wherein the table includes a listing of a not-permitted parent nodes for each child node.

18. (original) The method of claim 1, further comprising the additional steps of:

receiving a rule set identifying non-allowable combinations between child nodes and parent nodes in a hierarchical document structure; and

transforming the rule set into a table relating a class of parent nodes and a class of child nodes.

19. (original) The method of claim 1, further comprising the additional steps of:

receiving a rule set identifying non-allowable combinations between child nodes and parent nodes in a hierarchical document structure; and

transforming the rule set into a table relating parent nodes and child nodes.

20. (currently amended) A computer program product, tangibly encoded on a computer-readable medium, for searching for one or more logical elements in a hierarchical tree structure of an ~~extended~~extensible markup language (XML) document conforming to a schema used for XML,

comprising instructions operable to cause a programmable processor to:

provide a representation of an ~~extended markup language~~XML document instance containing two or more logical elements, wherein at least one logical element is a parent node and at least one logical element is a child node in a hierarchical tree structure describing the representation;

receive a query for logical elements satisfying an XPath expression; [[and]]

search in the hierarchical tree structure only nodes that potentially have child nodes satisfying the math expression; and

provide the logical elements satisfying the XPath expression.

21. (original) The computer program product of claim 20, wherein the instructions cause a programmable processor to generate a collection of parent nodes that potentially have child nodes satisfying the XPath expression from a table relating a class of parent nodes and a class of child nodes, and wherein the instructions cause the table to be used in the search.

22. (original) The computer program product of claim 20, wherein the instructions cause a programmable processor to generate a collection of parent nodes that potentially have child nodes satisfying the XPath expression from a table relating parent nodes and child nodes, and wherein the instructions cause the table to be used in the search.

23. (original) The computer program product of claim 21, wherein the table comprises entries containing hash representations of the class of parent nodes and class of child nodes.

24. (original) The computer program product of claim 22, wherein the table comprises entries containing hash representations of the parent nodes and child nodes.

25. (original) The computer program product of claim 22, wherein the table comprises a listing of a permitted class of child nodes for each class of parent node.

26. (original) The computer program product of claim 21, wherein the table comprises a listing of permitted child nodes for each parent node.

27. (original) The computer program product of claim 21, wherein the table comprises a listing of a permitted class of parent nodes for each class of child node.

28. (original) The computer program product of claim 22, wherein the table comprises a listing of permitted parent nodes for each child node.

29. (currently amended) The computer program product of claim 20, further comprising instructions to:

receive a rule set identifying allowable combinations between a class of child nodes and a class of parent nodes in a hierarchical document structure;

transform the rule set into a table relating the class of parent nodes and the class of child nodes; and use the table in the [[final]] search.

30. (currently amended) The computer program product of claim 20, further comprising instructions to:

receive a rule set identifying allowable combinations between child nodes and parent nodes in a hierarchical document structure;

transform the rule set into a table relating parent nodes and child nodes; and use the table in the [[final]] search.

31. (original) The computer program product of claim 29, wherein:

the rule set includes one of: an XML schema, a DTD, and a RelaxNg schema.

32. (original) The computer program product of claim 30, wherein:

the rule set includes one of: an XML schema, a DTD, and a RelaxNg schema.

33. (original) The computer program product of claim 21, wherein the wherein the table includes a listing of a class of not-permitted child nodes for each class of parent node.

34. (original) The computer program product of claim 22, wherein the wherein the table includes a listing of not-permitted child nodes for each parent node.

35. (canceled)

36. (canceled)

37. (original) The computer program product of claim 21, wherein the wherein the table includes a listing of a class of not-permitted parent nodes for each class of child node.

38. (original) The computer program product of claim 22, wherein the wherein the table includes a listing of not-permitted parent nodes for each child node.

39. (original) The computer program product of claim 20, further comprising instructions to:

receive a rule set identifying non-allowable combinations between a class of child nodes and a class of parent nodes in a hierarchical document structure; and

transform the rule set into a table relating the class of parent nodes and the class of child nodes.

40. (original) The computer program product of claim 20, further comprising instructions to:

receive a rule set identifying non-allowable combinations between child nodes and parent nodes in a hierarchical document structure; and

transform the rule set into a table relating parent nodes and child nodes.